REMARKS

Claims 2-10, 12-20 and 22-33 are pending in the present application. Claims 2, 12 and 22 have been amended, Claims 1, 11 and 21 have been cancelled, and Claims 31-33 have been added, herewith. Consideration/reconsideration of the pending claims is respectfully requested.

Amendments were made to the specification to correct errors and to clarify the specification. No new matter has been added by any of the amendments to the specification. Also, Applicants have submitted a replacement sheet for the drawing labeled FIG. 3.

Applicants would initially like to thank the Examiner for taking the time to conduct a telephonic interview on May 28, 2004. While no agreement was reached during such interview, distinctions between the claimed invention and the cited reference were discussed.

I. Specification

The Examiner commented on the proper language and format for an abstract of the disclosure. Applicants have amended the abstract herewith in an attempt to comply with the Examiner's comments

II. 35 U.S.C. § 102, Anticipation

The Examiner rejected Claims 1-30 under 35 U.S.C. § 102 as being anticipated by Blelloch et al. (US006434590B1). This rejection is respectfully traversed.

With respect to Claims 1, 11 and 21, such claims have been cancelled herewith without prejudice or disclaimer.

With respect to Claim 2 (and dependent Claims 3-10), such claim has been amended to be in independent form, and to clarify that the encapsulating step occurs as a part of the scheduling of the plurality of commands in programming order. These plurality of commands (which are scheduled for execution in a programming order) are selected from the environment in which commands (otherwise) execute concurrently. As to the rejection of Claim 2, Applicants show that the cited reference does not teach the claimed feature of *encapsulating*, after having been selected from the environment in

which commands otherwise execute concurrently, a first and second of the plurality of commands in a first and second process, respectively. This claimed feature advantageously provides the ability to sequentially execute commands which otherwise would execute concurrently.

The Blellock reference teaches a system having two subsets of processors – 'worker' processors which execute tasks and 'scheduler' processors which execute the scheduler (Col. 14, lines 4-42). All of the processing elements operate separately without being in synchronization (Col 5, lines 19-21). While it appears that the scheduler initially determines a sequential ordering of tasks for processing (Abstract 1-4), this sequential schedule is independent of the actual parallel execution that occurs (Abstract lines 10-15). The scheduled tasks are placed on a queue, where they are extracted for execution by the 'worker' processors (Col. 2, lines 43-51), thereby creating a pull system rather than a push system (Col. 8, lines 17-21). This independence between the scheduler and process execution is expressly desired to allow tasks to run in parallel and independent of one another (Col. 1, lines 36-48), and to improve overall system flexibility (Col. 5, lines 19-21).

While there are synchronization variables that can cause a given process thread to wait on another thread to set the synchronization variable (Col. 8, lines 43-59), these 'waiting' threads have already begun execution and are subsequently placed in a wait state, and are awakened when another process thread writes to the synchronization variable (Col. 10, lines 44-65). Thus, these synchronization variables do not teach otherwise suggest the claimed step of "beginning processing of said second process only in response to a completion of processing of said first process" as both the spawned thread and waiting thread have begun processing and thereby execute concurrently.

As to Blellock's task queue TQ1 (Fig 4), and even assuming that tasks which are placed on such queue may be sequentially executed, these tasks have not been encapsulated into a process after having been selected from an environment of tasks that otherwise execute concurrently. In other words, they are executed in the same task form as selected, without subsequent encapsulation, as claimed.

Thus, there is no teaching of scheduling execution of a plurality of commands in a programming order by encapsulating commands that have been selected from commands

that otherwise execute concurrently, such that a first one of the plurality of commands in said order begin and complete executing prior to a second one of said commands in said order beginning executing, wherein said plurality of commands are executed sequentially in said programming order.

Applicants traverse the rejection of Claims 12 (and dependent Claims 13-20) and 22 (and dependent Claims 23-30) for similar reasons to those given above regarding Claim 1.

Therefore, the rejection of claims 2-10, 12-20 and 22-30 under 35 U.S.C. § 102 has been overcome.

III. Newly Added Claims

Claims 31-33 have been added herewith. Examination is respectfully requested.

IV. Conclusion

It is respectfully urged that the subject application is patentable over the cited reference and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 6/10/04

Respectfully submitted,

Duke W. Yee
Reg. No. 34,285

Wayne P. Bailey
Reg. No. 34,289
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 367-2001
Attorneys for Applicants